



HAL
open science

A general framework for imprecise regression

Mathieu Serrurier, Henri Prade

► **To cite this version:**

Mathieu Serrurier, Henri Prade. A general framework for imprecise regression. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007), Jul 2007, London, United Kingdom. pp.1597–1602, <10.1109/FUZZY.2007.4295605>. <hal-03358862>

HAL Id: hal-03358862

<https://hal.science/hal-03358862v1>

Submitted on 30 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

A general framework for imprecise regression

Mathieu Serrurier and Henri Prade

Abstract—Many studies on machine learning, and more specifically on regression, focus on the search for a precise model, when precise data are available. Therefore, it is well-known that the model thus found may not exactly describe the target concept, due to the existence of learning bias. In order to overcome the problem of too much illusionary precise models, this paper provides a general framework for imprecise regression from non-fuzzy input and output data. The goal of imprecise regression is to find a model that has the better trade-off between faithfulness w.r.t. data and (meaningful) precision. We propose an algorithm based on simulated annealing for linear and non-linear imprecise regression with triangular and trapezoidal fuzzy sets. This approach is compared with the different fuzzy regression frameworks, especially with possibilistic regression. Experiments on an environmental database show promising results.

I. INTRODUCTION

Machine learning aims at building models that describe concepts from data. If the concept to be learnt consists in being able to assign one of a finite number of classes to an object, the problem is referred to classification. If the concept takes the form of a continuous function, we face a regression problem. The models learnt in regression are precise and considered as certain, in the sense that for given input values, the models provide a unique output value. Dealing with uncertainty and imprecision in learning is a research field which is more and more explored. Classical regression has been initially extended to fuzzy data by adapting the least square fitting criterion to the fuzzy case [4] (see also [14], [18], [1], [5] for overviews and general discussions). Therefore, these approaches handle fuzzy inputs and/or fuzzy outputs. The interest of this kind of approach is limited to the cases where fuzzy data are naturally found. A possibilistic regression method has also been proposed [17]; its aim is to learn an imprecise model even when the data are non-fuzzy. But this approach is not fully satisfactory since it is very sensible to outliers and provides a representation of imprecision based only on intervals. Moreover, it does not take into account the imprecision that is naturally associated with machine learning bias.

Integration of imprecision in regression models would allow us to take into account both the incompleteness of the information provided by the data and the limitations of the descriptive power of the model. From the formal machine learning point of view, we know that the incompleteness of the information provided by the data and the limitations of

the descriptive power of the model implies a boundary on the accuracy of the learning process. Thus, learning a model that is certain and precise may be illusionary since this model will necessarily be false. When learning imprecise models, we overcome this limitation by describing not only the general tendency of the data, but also the possible variation around it. The general idea is to find the most precise model compatible with the data and the learning bias.

In this paper we propose a general framework for imprecise regression. Given non-fuzzy input and output data, the goal is to find a model as precise as possible that provides the most faithful description of the data. Properties needed for precision evaluation functions are defined. Next, we suggest a candidate precision function and we explicitly define it for trapezoidal and triangular fuzzy sets. We describe linear and non-linear imprecise functions for this kind of fuzzy sets. Since finding the global optimum is not tractable for a deterministic algorithm in this context, a simulated annealing approach is presented. We then discuss the difference between imprecise regression and the other types of fuzzy regressions, especially with the possibilistic one. Lastly, we apply the simulated annealing approach to an environmental dataset.

II. A GENERAL FRAMEWORK FOR IMPRECISE REGRESSION

When learning from crisp (non-fuzzy) data, we can consider two important kinds of biases. The first one comes from the data description itself. More precisely, the language used constitutes a bound for the descriptive power of the data and leads to an incomplete view of the world. This aspect is reinforced by the existence of noise that may appear when collecting the data and by the consideration of only a finite set of examples. The other major bias is the complexity of the hypothesis space. Indeed, due to the limitation of the hypothesis language and the complexity of the algorithms, it is rarely possible to find the hypothesis that describes exactly the concept we want to learn. One of the major machine learning theorems [19] shows that these biases lead to a bound on the effectiveness of the learning method used. In this context, learning a crisp and certain model is illusionary since this model corresponds to a fake view of the world. It is especially obvious when considering linear crisp regression. The bias on the data appears first in the consideration only of a sample set of data, and in the input variables that may not be sufficient for describing the concept (for instance, two examples may have the same values for input variables and different values for output ones). In the same way, using the hypothesis space of

Mathieu Serrurier, DSNA/R&D, 7 avenue Edouard Beulin, 31055 Toulouse, France E-mail: serrurie@cena.fr

Henri Prade, IRIT, University of Toulouse III, 31062 Toulouse Cedex 9, France, E-mail: prade@irit.fr

linear functions for algorithmic simplicity, presumes that the concept to be learnt is linear too, which is generally not the case. Thus, the function learnt is arbitrarily precise and does not take into account the imprecision entailed by the data and the hypothesis bias.

The goal of a general framework for imprecise regression is to overcome the learning biases by considering them as factors that have impact on the precision of the models rather than as a boundary to the effectiveness of the learning process. Knowing that the representation of the examples and the hypothesis correspond necessarily to an incomplete view of the world, we will search for imprecise hypotheses that take into account this incompleteness. Imprecise hypotheses can then describe in a more realistic and faithful way the concept instead of learning arbitrary precise hypotheses. Thus, given a set of crisp data, we will search the model that is as precise as possible and which provides the best description of the data. When the imprecision tends to 0, we obtain a crisp hypothesis that describes the concept exactly. In a formal way, imprecise regression allows us to represent the imprecision associated with the model by taking into account the incompleteness of the information provided by the data and the representation of the hypotheses.

The problem can be stated as follows. A regression data is a set of m pairs (\vec{x}_i, y_i) , $1 \leq i \leq m$, where $\vec{x}_i \in \mathcal{X}$ is a vector of n input variables and $y_i \in \mathbb{R}$ is the real output variable. An imprecise function F is a function from \mathcal{X} to $(\mathbb{R} \rightarrow [0, 1])$ that associates a distribution on the possible values of the output to the inputs vector \vec{x} . This distribution may be described by a fuzzy set, a possibility distribution or a probability distribution. In the following, we only consider normalized possibility distributions, which can be viewed as a fuzzy set. This will be denoted by $F(\vec{x}_i) = \pi_i$. Since the goal of imprecise regression is to find a function, as precise as possible, which describes the concept to be learnt, we first need to define the evaluation function of the precision of a possibility distribution. The precision of a fuzzy set is an information theory issue, and several measures have been proposed. Intuitively, the information provided by a possibility distribution decreases with the surface of the area under it. Let us first define the surface of a possibility distribution.

Definition 1 (Possibility distribution surface): Given $\pi : \mathbb{R} \rightarrow [0, 1]$ a possibility distribution under π , the surface $S(\pi)$ of this function is given by :

$$S(\pi) = \int_0^1 l(\pi_\alpha) d\alpha \quad (1)$$

with for $\alpha > 0$

$$l(\pi_\alpha) = \mu(\{y \in \mathbb{R}, \pi(y) \geq \alpha\})$$

where μ is the Lesbegue measure and

$$l(\pi_0) = \lim_{\alpha \rightarrow 0} l(\pi_\alpha).$$

In the spirit of [21], we restate general requirement for a precision measure.

Definition 2 (Precision measure properties): Given $\pi : \mathbb{R} \rightarrow [0, 1]$ a normalized possibility distribution and Pr a precision function :

- 1) Pr is a function from $(\mathbb{R} \rightarrow [0, 1])$ to $[0, 1]$
- 2) $Pr(\pi) = 1$ iff $\exists y \in \mathbb{R}$ such that $\pi(y) = 1$ and $\forall y' \in \mathbb{R}, y' \neq y, \pi(y') = 0$.
- 3) $Pr(\pi) = 0$ iff $\forall y \in \mathbb{R}, \pi(y) = 1$.
- 4) if $S(\pi') \geq S(\pi)$ iff $Pr(\pi') \leq Pr(\pi)$.

The first condition expresses that the precision of a fuzzy set is bounded by $[0, 1]$. Conditions 2 and 3 assure that the most precise possibility distribution is a Dirac (with a null surface) and the less precise distribution is the one that gives the same possibility level to all elements in \mathbb{R} . The last condition states that the precision must decrease when the surface of the possibility distribution increases. Note that since the surface under a probability distribution is constant, the above conditions should be adapted in order to describe uncertain regression. Let us now define the goal of imprecise regression.

Definition 3 (imprecise regression): Given a data set made of m pairs (\vec{x}_i, y_i) , $1 \leq i \leq m$, and the precision function Pr the goal of imprecise regression is to find the function F that maximizes

$$R(F) = \frac{\sum_{i=1}^m \pi_i(y_i) * Pr(\pi_i)}{m} \quad (2)$$

with $\pi_i = F(\vec{x}_i)$.

By maximizing together the accuracy of the imprecise function, here estimated through the terms $\pi_i(y_i)$'s, and its precision, we ensure the best tradeoff between accuracy and precision of the model. Maximum is reached when the function describes exactly the data. In this case, we obtain a crisp regression. Since the learning bias may prevent reaching this maximum, the function will describe both the general tendency of the data and the variations around it. Of course, the quality of the tradeoff depends on the choice of a precision evaluation function and of its scalability with the data.

III. A SIMULATED ANNEALING APPROACH FOR IMPRECISE REGRESSION

A. Notations

In this section, we propose an algorithm for imprecise regression where the imprecise function to be learnt has an output that is assumed to be a triangular or a trapezoidal fuzzy set. Let us first introduce some notations. Triangular-shaped fuzzy sets are defined as follows :

$$I_{l,m,r}(x) = \begin{cases} 0 & \text{if } x \leq l \text{ or } x \geq r \\ \frac{x-l}{m-l}, & \text{if } x \leq m \text{ and } x > l \\ \frac{r-x}{r-m} & \text{if } x > m \text{ and } x < r \end{cases}$$

Trapezoidal fuzzy sets are encoded as this :

$$T_{a,b,c,d} = \begin{cases} 0 & \text{if } x \leq a \text{ or } x \geq d \\ 1 & \text{if } x \geq b \text{ or } x \leq c \\ \frac{x-b}{b-a} & \text{if } x < b \text{ and } x > a \\ \frac{d-x}{d-c} & \text{if } x > c \text{ and } x < d \end{cases}$$

Yager [21] has proposed a specificity function for a possibility distribution π defined on \mathbb{N} :

$$Pr_{\mathbb{N}}(\pi) = \int_0^1 \frac{1}{|\pi_\alpha|} d\alpha$$

with $|\pi_\alpha| = \{y \in \mathbb{N}, \pi(y) \geq \alpha\}$. The problem is that this function is not defined on \mathbb{R} . Thus, we need to adapt this function to continuums. Moreover, since the output range of the data may vary highly, we introduce a normalization constant. Then, the precision function used will be the following :

$$Pr(\pi) = \int_0^1 \frac{1}{1 + C * l(\pi_\alpha)} d\alpha$$

with a constant $C \in \mathbb{R}$. More precisely, when applying this function to triangular or trapezoidal fuzzy sets we obtain :

$$Pr(I_{l,m,r}) = \frac{\ln(1 + C * (r - l))}{C * (r - l)}$$

and

$$Pr(T_{a,b,c,d}) = \frac{\ln(1 + C * (b - a + d - c) + C * (c - b))}{C * (b - a + d - c)}.$$

We will consider two different types of imprecise functions. An fuzzy-valued function $F_{l,m,r}$ that associates triangular fuzzy sets to inputs \vec{x} is represented by three scalar functions (f_l, f_m, f_r) that describe respectively the three parameters of the fuzzy set. Thus we have :

$$F_{l,m,r}(\vec{x}) = \langle f_l, f_m, f_r \rangle (\vec{x}) = I_{f_l(\vec{x}), f_m(\vec{x}), f_r(\vec{x})}.$$

In the same way, an fuzzy-valued function $F_{a,b,c,d}$ that associates trapezoidal fuzzy sets to inputs \vec{x} is described as follows :

$$\begin{aligned} F_{a,b,c,d}(\vec{x}) &= \langle f_a, f_b, f_c, f_d \rangle (\vec{x}) \\ &= T_{f_a(\vec{x}), f_b(\vec{x}), f_c(\vec{x}), f_d(\vec{x})}. \end{aligned}$$

The different scalar functions may be taken for instance as linear functions of the forms

$$f(\vec{x}) = \langle x_1, \dots, x_n \rangle = a_0 + a_1 * x_1 + \dots + a_n * x_n$$

or may be given by neural networks. When using linear functions, finding optimal f_l, f_m, f_r or f_a, f_b, f_c, f_d constitutes a hard problem which is not solvable by classical optimization methods. In the same way, the back-propagation algorithm is not applicable in this case since we need to optimize three or four neural networks with one target quality function. We propose to solve the problem by using a simulated annealing algorithm.

Alg. 1 Simulated annealing(V, E, X_0, T_0, T_{min})

```

1:  $X = X_0$ 
2:  $T = T_0$ 
3: while  $T < T_{min}$  do
4:    $Y =$  random element in  $V(X)$ 
5:    $dE = E(X) - E(Y)$ 
6:   if  $dE > 0$  then
7:      $P = e^{-\frac{dE}{T}}$ 
8:      $A =$  random value in  $[0, 1]$ 
9:     if  $A < P$  then
10:        $X = Y$ 
11:     end if
12:   else
13:      $X = Y$ 
14:   end if
15:   decrease the value of  $T$ 
16: end while
17: return  $X$ 

```

B. Algorithm

1) *Simulated annealing*: Simulated annealing [15] is a meta-heuristic method developed for optimization problems. The general form of simulated annealing algorithm is as in Alg. 1. This method is inspired from a well-known physical phenomenon, coming from metallurgy. Let us consider a function $E : s \mapsto \mathbb{R}$ to be minimized, and representing the energy of a statistical mechanical system in a given state s . The probability for the system to go from the state s to the state s' at the temperature T is given by the Boltzman-Gibbs distribution $P(s) = e^{-\frac{(E(s') - E(s))}{kT}}$ where k is the Boltzmann constant (in Alg. 1. the constant k is merged with the parameter T). For high values of T , all states have a high probability to be accepted. On the opposite side, when T is close to zero, only states improving the current minimization of the function will be accepted. The convergence to the global optima is granted when using a logarithmic decrease of the temperature. Given an initial temperature T_0 , the temperature at step t of the algorithm is $T_t = r * T_{t-1}$ for $t \geq 1$ and $0 < r < 1$. With this method, we are not sure to find the global minimum, but we can expect to find at least a local one better than the one computed by a greedy algorithm. The values of temperatures T_0 , T_{min} , and r are critical and depend on the problem we want to solve. The initial state can be chosen randomly, since a "good" initial state will be "forgotten" at a high temperature.

2) *Application to imprecise regression*: As pointed out previously, classical optimization approaches or back-propagation are not usable in this context. Thus, we use a simulated annealing approach. Then, the goal is to determine the function F that maximizes equation 2 in the space of the imprecise functions. In order to use simulated annealing, we first need to define the neighborhood V of a function F . For the function of the form $F_{l,m,r}$ and $F_{a,b,c,d}$ we define

neighborhood as follows :

$$V(F_{l,m,r}) = \langle V(f_l), V(f_m), V(f_r) \rangle$$

and

$$V(F_{a,b,c,d}) = \langle V(f_a), V(f_b), V(f_c), V(f_d) \rangle .$$

The neighborhood of a linear function is obtained by randomly adding or removing a fixed small value to all the coefficients. In the same manner, neighborhood of a neural network is computed by randomly adding or removing a fixed small value to every weight that links the different nodes of the networks. The use of a fixed small value for the variations is due to the fact that simulated annealing is designed for discrete exploration of the space. Given the type of fuzzy-valued function wanted (triangular or trapezoidal-shaped), parametrized in a way this is linear or not, the corresponding neighborhood is used in the simulated annealing algorithm.

IV. RELATED WORKS

In this section, we emphasize the differences between the imprecise regression approach presented in this paper and the other approaches to fuzzy regression. Two types of approaches can be distinguished : fuzzy least square regression and possibilistic regression.

The first type of fuzzy regression deals with fuzzy output data, and maybe fuzzy inputs, which have to be described by a fuzzy regression function. Such a method has been proposed for dealing with imprecise data by several authors [3], [4]. The Diamond's method is based on the extension of the L_2 metrics to fuzzy sets. Thus, once a distance between fuzzy sets is defined, the method used for fuzzy regression is the least square error minimization. The problem has been solved in [4] for linear regression with fuzzy inputs and/or fuzzy output. This approach has been extensively studied in the linear case [20], [7], [8], [16]. Non-linear approaches have also been proposed by using for instance neural networks [6], [11], SVM's [10] and genetic algorithms [2]. The major advantage of the least square method is that it is the natural mathematical extension of the crisp regression. In this context, when data inputs and output are not fuzzy, fuzzy least square regression is equivalent to classical least square regression (even if crisp data are viewed as particular fuzzy numbers). This constitutes the principal difference with our approach. In fact, imprecise regression aims at representing the imprecision associated with the model. This contrast with fuzzy least square regression that deals with the imprecision/fuzziness of the data.

The second type of approach, named possibilistic regression, was initially proposed by Tanaka [17]. The goal of this approach is to associate the data with a pair of upper and lower regression function, while minimizing the total spread of the output. In order to do that, a lower bound and an higher bound of the regression function are computed. This method can be used with crisp data and/or with fuzzy

data. A linear model [17] has been initially proposed. In the first step, the linear regression function that produces the interval is learned by solving a linear programming problem. Then, an interval-valued linear regression function is obtained. When dealing with crisp data, this function associates an interval (rather than a crisp value) to the input. However, this method can be very costly when dealing with a large number of input variables. This method has been extended to non-linear possibilistic regression in [12] by using neural networks. In this case, lower bound and higher bound functions are represented by two different neural networks. The advantage of possibilistic regression with respect to the first type of fuzzy regression is that it can handle both crisp data and fuzzy data. The main disadvantage of this method is that it is very sensitive to outliers. In fact, the optimal upper bound function, for instance, is the function that is immediately above the whole set of data. Thus, outliers may affect to a large extent the function that is learnt. This drawback has been taken into consideration in [13] by combining SVM's with neural networks in order to decrease the sensitivity to outliers.

At first glance, imprecise regression may seem to be very close to possibilistic regression. First, the two approaches deal with crisp data. Second, they use separate functions, described by a linear function or a neural network, in order to represent fuzzy sets or intervals. However, the two approaches differ both at the theoretical level and at the algorithmic level. Possibilistic regression aims at finding the most precise function that is totally accurate with respect to all the examples. On the contrary, the goal of imprecise regression is to find the function that has the better tradeoff between faithfulness and precision in order to take into account biases associated with the learning problem. This is why imprecise regression is less sensitive to outliers than possibilistic regression. Moreover, lower bound and higher bound of the function in possibilistic regression are learned separately. Thus, when dealing with crisp data, possibilistic regression can only produce intervals rather than genuine fuzzy sets, and there is no guarantee of the coherence between the higher and the lower bound of the interval. Imprecise regression quality measure is global, and all the functions that describe the fuzzy sets are learnt together. It allows us to learn models that can represent the imprecision by any kind of fuzzy sets (here triangular and trapezoidal) in a coherent way.

V. EXPERIMENTATIONS

The problem considered here, for illustrative purpose, is to learn the concentration of a polluting agent in a water spring. Imprecise regression is used since it handles crisp inputs and outputs. The main benefit of imprecise regression with respect to classical regression is that imprecise regression both describes the general tendency of the function (as classical regression does) and the amount of variation around the general tendency. In this application, allowing for imprecise models enables the user to know to what extent the

prediction can be made in a precise way. Thus a fuzzy set that expresses the uncertainty of the prediction provides a more valuable piece of information for a comparison with a reference threshold (for firing alerts for instance). The data describe concentration of atrazine pollutant with respect to the time. Data contain more than 300 pairs of input-output pieces of data. We use the algorithm described in section 3 with linear and non-linear functions for triangular and trapezoidal fuzzy sets. The constant C is fixed such as $C = \frac{1}{2+\sigma}$ where σ is the variance of the output data. Roughly speaking, it means that outputs are considered as really relevant when they do not differ from the global tendency more than σ . The higher C , the narrower the fuzzy function. Neural networks have one input node and five hidden nodes. Initial temperature of the simulated annealing is fixed at 0.005 and ratio of the geometric scheduler is equal to 0.9995. The different functions learnt are represented on Figures 2 to 5, Figure 1 corresponding to classical linear least square regression. Fuzzy sets are represented in the third dimension. The following table provides the numerical results.

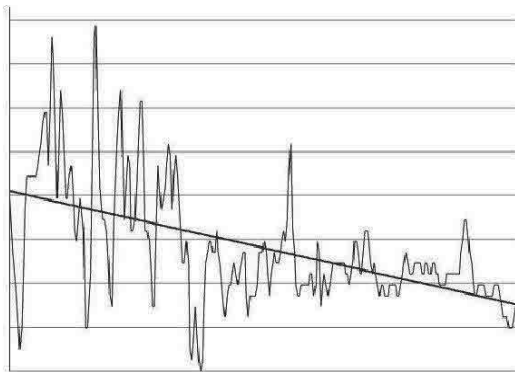


Fig. 1.

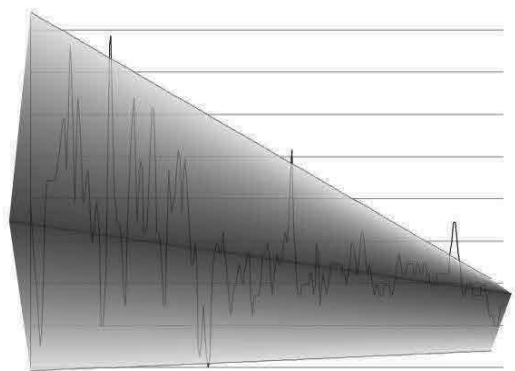


Fig. 2.

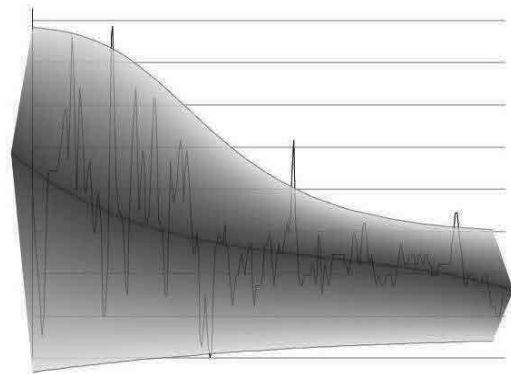


Fig. 3.

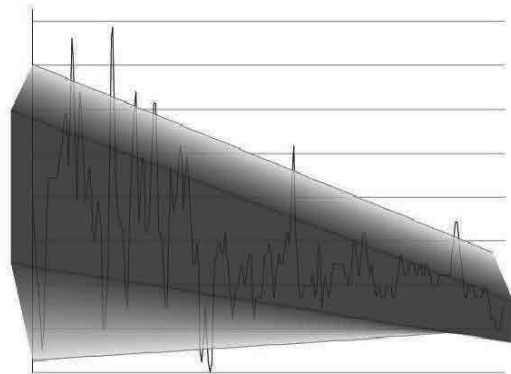


Fig. 4.

Several comments are in order. First, Figure 1, that represents classical regression, shows that concentration globally decreases with time. However, this is not fully satisfactory. In fact, by using linear function, we cannot expect to represent the data exactly. Moreover, the classical linear function does not give any clue of how imprecise is the result. Imprecise regression allows us to have this information. Regardless the consideration of non-linear or linear fuzzy regression with triangular or trapezoidal fuzzy sets, the general tendency remains the same as for the classical case. It also shows that variation around the global tendency decreases with time. In all the cases, we can observe that fuzzy sets are not symmetrical and this indicates that variations more often correspond to a decrease of the output. In the case of pollutant prediction, this is crucial because it shows that variations are smaller and smaller with time and then the plausibility to overcome a risk threshold will be easier to predict. When considering results more precisely, we observe that non-linear settings (Fig. 3 and 5) are more accurate than linear ones (Fig. 2 and 4). Triangular shaped fuzzy sets allow us to distinguish directly the general tendency by considering the top of the triangle. Trapezoid fuzzy sets give a more precise description of the variation around the general tendency.

Observations derived from the Figures are confirmed when looking at the above result table. As expected, it shows two

Type	avg. member. deg.	avg prec.	R(F)
lin. triang.	0.706	0.586	0.417
non lin. triang.	0.712	0.593	0.427
lin. trap.	0.871	0.492	0.434
non lin. trap.	0.866	0.501	0.439

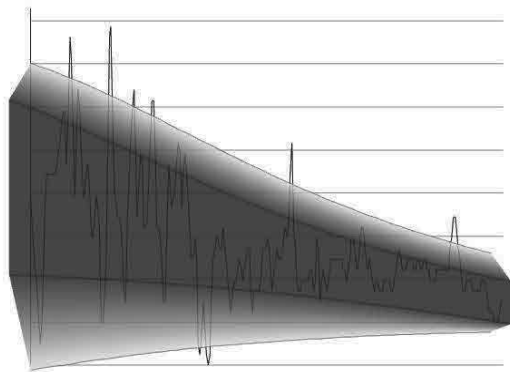


Fig. 5.

things. First, non-linear models give better results. It is due to the fact that non-linear functions have a better representation power than linear ones while having a smaller learning bias. Second, although trapezoidal fuzzy set functions have a smaller precision result than triangular ones, they yield better results for the evaluation function R since they are more accurate. This is not a surprise since triangular fuzzy sets can be viewed as a particular kind of trapezoidal fuzzy sets. Finally, the results of the experimentations agree with the goal of the general framework of imprecise regression. It learns the most precise function that describes the data. As expected, what is obtained is a description of the general tendency of the data together with the variation around it. Moreover, results show that imprecise regression is not too much sensitive to outliers. Finally, the precision of the learnt function is related to the expressive power of the hypothesis space, and then to the learning biases associated to the models.

VI. CONCLUSION

In this paper, we have proposed a new kind of fuzzy regression. Imprecise regression aims at learning fuzzy functions from crisp input and output data. The goal of the imprecise regression is to overcome the learning biases, that make precise regression illusory, by learning together the general tendency of the data and the variation around it. In this context, we have presented a general framework that defines imprecise regression as a search for the best tradeoff between accuracy and precision. Given a fixed precision evaluation function, the global solution is not tractable with deterministic algorithms. Thus, we describe a simulated annealing-based algorithm for learning triangular and trapezoidal-shaped linear and non-linear fuzzy functions in the imprecise regression framework. We pointed out that although imprecise regression has some similarities with possibilistic regression, it seems to be more powerful and more adapted to regression problems with crisp inputs and output data. Especially, it is less sensitive to outliers and allows us in principle to learn any kind of fuzzy functions. Moreover, preliminary experimentations also indicate promising results.

In the future, the general framework of imprecise regression should be more thoroughly studied. Especially, it should be interesting to explore the links between learning biases and optimal imprecise functions. More experimentations will also be useful for comparing possibilistic regression and imprecise regression and for testing different kinds of precision evaluation functions. Currently, we work on the application of imprecise regression to the learning of performance measures for a pointing task known as Fitts' task [9] in Human-Machine interaction.

REFERENCES

- [1] H. Bandemer and W. Naether. *Fuzzy Data Analysis*. Kluwer Academic Publishers, Dordrecht, Boston, London 1992, 1992.
- [2] J.J. Buckley and T. Feuring. Linear and non-linear fuzzy regression : Evolutionary algorithm solutions. *Fuzzy Sets and Systems*, 112:381–394, 2000.
- [3] A. Celmins. Least squares model fitting to fuzzy vector data. *Fuzzy Sets Syst.*, 22(3):245–269, 1987.
- [4] P. Diamond. Fuzzy least squares. *Inf. Sci.*, 46(3):141–157, 1988.
- [5] P. Diamond and H. Tanaka. Fuzzy regression analysis. *Fuzzy Sets in Decision Analysis, Operations Research and Statistics. The Handbooks of Fuzzy Sets Series*. Kluwer Academic Publishers, pages 349–387, 1998.
- [6] J. P. Duniak and D. Wünsche. Fuzzy regression by fuzzy number neural networks. *Fuzzy Sets Syst.*, 112(3):371–380, 2000.
- [7] P. D'Urso. An "orderwise" polynomial regression procedure for fuzzy data. *Fuzzy Sets and Systems*, 130:1–19, 2002.
- [8] P. D'Urso. Linear regression analysis for fuzzy/crisp input and fuzzy/crisp output data. *Comput. Stat. Data Anal.*, 42(1-2):47–72, 2003.
- [9] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.
- [10] D. H. Hong and C. Hwang. Support vector fuzzy regression machines. *Fuzzy Sets Syst.*, 138(2):271–281, 2003.
- [11] H. Ishibuchi and M. Nii. Fuzzy regression using asymmetric fuzzy coefficients and fuzzified neural networks. *Fuzzy Sets Syst.*, 119(2):273–290, 2001.
- [12] H. Ishibuchi and H. Tanaka. Fuzzy regression analysis using neural networks. *Fuzzy Sets and Systems*, 50:57–65, 1992.
- [13] J. T. Jenga, C. C. Chuang, and S. F. Su. Support vector interval regression networks for interval regression analysis. *Fuzzy Sets and Systems*, 138 (2):283–300, 2003.
- [14] J. Kacprzyk and M. Fedrizzi (Eds.). *Fuzzy Regression Analysis*. Omnitech Press, Dordrecht, Boston, London 1992, 1992.
- [15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [16] M. Modarres, E. Nasrabadi, and M. M. Nasrabadi. Fuzzy linear regression models with least square errors. *Applied Mathematics and Computation*, 15:873–881, 2003.
- [17] H. Tanaka. Fuzzy data analysis by possibilistic linear models. *Fuzzy Sets Syst.*, 24(3):363–376, 1987.
- [18] H. Tanaka and P.J. Guo. *Possibilistic Data Analysis for Operations Research*. Studies in Fuzziness and Soft Computing, Vol. 29, Warsaw, and Physica, Heidelberg, 1999.
- [19] V.N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- [20] H.-F. Wang and R.-C. Tsaur. Resolution of fuzzy regression model. *European Journal of Operational Research*, 26:637–650, 2000.
- [21] R. Yager. Measuring tranquility and anxiety in decision making: An application of fuzzy sets. *International Journal of General Systems*, 8:139–146, 1982.